IJEPS 3 (2): 22-36 (2020)

Control traffic balancing optimization for in-band software defined networks

Alaaedi Hussain, Masoud Sabaei

Amirkabir University of Technology, Iran
Corresponding Author Email: halaaedi@aut.ac.ir, sabaei@aut.ac.ir

In Software Defined Networking (SDN), when a packet arrives from a new flow to software defined switch, a control message must be sent from this switch to the central controller. This control message has to be sent as fast as possible, and when it reaches the central controller, it must response it quickly because the switch will still be waiting until receives the response from the controller which tells the switch how to route the packets of the new flow. In largescale SDN, where there are thousands of flows arrive every second, significant amount of these control messages will be generated and forwarded in the network. This huge amount of control messages becomes a problem especially in in-band control channel. Since, most traffic balancing works focused on data traffic balancing. This paper tries to give attention to the control traffic balancing considering data traffic in the same time. This problem is very important for SDN because the delay time of the control message directly affects the overall performance of SDN. A non-convex optimization problem is presented to balance the control traffic over the network links. Simulation results shows that the proposed model can balance control traffic more efficient than other models and needs less on-line solving time to find near optimal routes.

Keywords: Software defined networking, SDN, control traffic, load balancing, In-Band Control

1. Introduction

The Software Defined Networks (SDNs) have been realized as networking paradigm of next-generation by vow for demonstratively provide the resource usage of network, easily the management of network, decline the cost of operating, also raise the assessment, innovation [1-4]. A main opinion on SDNs is abstracting plane of data from control plane with:

1. Eliminating the decisions of control of forwarding hardware, such as, switchers/routers.
2. Empowering forwarding hardware for being programmable via standardized, open intervene, such as, Open flow [2].
3. Utilizing the network controller by protecting the applications of management (like resource allocation, routing) for explaining (in software) the forwarding infrastructure of network manner.

By those separating architecture of networking, messages of control timely delivery to every software explained/ the Open flow greatly switch effects on SDNs effective, efficiency, particularly if it is in-band status [5] is utilized to control traffic which can outstandingly impacts performance of system by composed traffic of main information, messages of control. Therefore, this will become the large challenge for protecting in-band control traffic load-balancing to minimum delay of network through the controller which is concentrated inSDNs. In addition, a lot of activities of control plane, signalling events are needed in the SDNs [3, 4, 6]. Also they can simply produce the outstanding control traffic value which have to be mapped jointly by traffic of information. In another way, this will be cost-expensive for developing the great-scale of SDN by the out-band control that every switch will be joined straight to controller by the isolated channel of control. Different of the traffic balancing of data that purpose for dividing traffic of data evenly flows between the joints of network, traffic balancing of control is more challenging, special to the in-band control [4]. This purpose for obtaining forwarding paths of every switch control messages in those path which control the delay of message could be reduced the topic for adaptable performance to main traffic of data. The issue of control traffic forwarding is widely critical in the SDNs due to control traffic deliver timely triggered with the switches of open flow, such as, each innovative flow first package, state of accumulation/traffic, straight effects strategies of routing effective distinguished with controller. This target for obtaining every switch forwarding paths message of control in these ways which control the delay of message could be reduced topic for the performance which is adaptable to main traffic of data. With utilizing the ranking theory of network [7], writers in [1] mapped problem of control traffic forwarding with formulate this as the optimization issue which is non-linear. Although, this formulation complexity was largely high because of:

1. This is non-linearity.
2. Large link traffic allocations reliefs to the networks in great-size.

Thereby, they attempt for obtaining the near, close solution which is optimal in space of solution [8]. To that, firstly they assayed balancing issue of control traffic basic structure with prove the complexity of polynomial time, like, the polynomiality [8]. Particularly, issue of optimization was explained as the hardly convex framework, also this is proved that answer is able to be neared with the polynomial-time rapid algorithm. Specially, they also offered the polynomial-time approximation algorithm (PTAA) which uses ADMM attitudes update laws of primal-dual for solving great- scale convex optimization issue which is formulated. Specially, the PTAA is the

repeated algorithm which accuracy approaches answer which is optimal by the convergence that is rapid. Also they proved that the PTAA pursues rate of convergence that is fast O(1/cm) by the stable c>1, iteration amount m. The proposed algorithm in [1] does not obtain the optimal solution, but a near optimal solution. In our work, using the results obtained from [1], we will define new optimization to obtain the same solution in relatively shorter time. We use aim to improve the work [1] in two directions:

a) We consider individual delay for each link in the network instead of considering average network delay which leads to more load balancing on the network.
b) Considering that the long solving time comes from the non-linear nature of the problem, we try to reformulate the non-linear terms in order to decrease solution space which may decrease solution time. Rest of this paper is managed as below. In part 2, related work, and the paper motion are provided. Model of system is provided in part 3. The issue of traffic balancing control and basic issue structure are considered in part 4. Assessment of performance is provisioned in part 5, this paper is deducted in part 6.

2. Related Works

In this section, we examine several strategies of load balancing for the SDN that is the network infrastructure which is advanced. Many existence methods of network infrastructure to the load balance is not proper to the SDN that network components central controller is created with the planning section. This is simple for controlling the load to the server is joined is network infrastructure controller section physically. However, load balancing for the servers on SDN network needs encasing the different standards the server management of server. In [9], the author accents importance to the data servers– the centers with publishing needs that income along some alike servers of web. This summarize on recent mechanism of load balancing on the SDN. Network entrance cases demand the presented servers of load balancing according to stabilize load on every network case on infrastructure of SDN. This watched that load of network load always firm, resistance. This is saw that load balancing is performed on the server on sever congestion based situation. However, this does not contain congestion on channel. The solution offered for uniting the mechanisms of SDN for executing load balancing according to simplify problems attitude by recent system. The operation provisioned main demand recognizing to the load balancing of server, flaws which are existence on existence design of load balancing. In the operation, also this was offered for moving to the methods of load-aware server load balancing to the infrastructure of network which is advanced similar to SDN. Author in [10], discussed about SDN architecture, the custom architecture of network advantageous. In [11], different innovations were provisioned that is able to be utilized by networks of Software-Defined. Active load balancing is indexed as feasible innovation of later-generation in infrastructure of network that is advanced. In [12], the white paper on the SDN with foundation of Open Networking, the networking efficiency feasible by the SDN, is argued in how, also length SDN divers of custom devices of networking is noted. SDN feasible usage points are considered. Operation demonstrated in [13] provisioned demand to the specialized module of hardware to the balancers of load is removed by Open-flow techniques usage. So, demand to the specialized load balancer of load in every entrance is removed by centralized controller usage. Former, they explained hybrid controller usages, every to balancing the special application type, mails of statement, memory, browsers and so on. To have the sole algorithm of balancing optimal-fit to every application. In paper [14] founder by Zhihao Shang and his

colleagues, specialized module of load balancing hardware demand is removed with offering dynamic load balancing usage techniques by utilizing the Open Flow. Hai long and his colleagues in [15] compared dynamic, static techniques of load balancing performance with 3 tests name priority ratio, round-robin techniques. In technique of round-robin, the server successively hands over needs on demands number/time duration-based. Server obtains load on traffic pre-explained ratio-based on software-explained infrastructure of network. When load in basic server becomes very heavy for handling, so server of backup is empowered for handling load of traffic on load balancing techniques based on the priority. Operation [16] discusses that the load-balancing comes effective just when balancing done later of considering the server states of health. Parameters indexed below the server state of health are server physical resources state, also applications accessible state on server. An answer is offered is the EHLBOF on the POX controller. System timely searches servers for obtaining parameters of health state. At the point each server means the sickness, so server state is upgraded for below, later need wont sent to the server. Operation [17] offered that for utilizing Round-robin (DNS based) technique to the load balancing, balancing servers later of considering recent load, services performance metrics would be much more effective, ideal. In [18] founder by Deepika and his colleagues, dynamic, static techniques of load balancing are examined for obtaining the two systems pros and cons. Operation [19] discusses the technique for achieving the traffic-aware dynamic load balancing. This utilizes of 4 various modules: module of traffic discovery, module of load calculation, module of dynamic load scheduling, module of flow management. Authors in [20] discusses that the dynamic Open Flow empowered the strategy of load balancing executes more optimal than custom techniques such as Round-robin technique. The tests ascertain that the dynamic technique LABERIO out-executes the Round Robin with declining transferring time with 13 percent. In [21], requests grouped in CPU concentrated, storage concentrated, IO concentrated. After that, requests are determined to proper server that has resources that are the most un-utilized ones in order to requests progressing concentration. Operation [22] provisioned he techniques for balancing hybrid machines which are virtual and physical. VMs and PMs outright usage like CPU, storage, also network is checked, current assignment of resource is done on monitored parameters-based. In [23], they offer that to the efficiency in immediately entrance discovery system speed, parallel holding of the data with hybrid servers is the dream selection. Also this is derived that the system of load balancing is able to be utilized to parallel holding of data. The problem of control traffic balancing was considered in [1]. According to reduce middle delay in network, they explained the issue which is non-convex for obtaining an optimal path of every for controller. Particularly, fundamental mathematical structures which are fundamental of an issue of formulated non-linear, answer group were provisioned, so, the algorithm that was effective, known as polynomial-time approximation algorithm (PTAA), was offered for give rapid convergence for the close optimal answer with applying the alternating direction method of multipliers (ADMM). Although the proposed PTAA achieves considerable delay reduction, it has some shortcomings and defects which we try to overcome in the paper. One defect is that the objective function is defined to minimize the average delay in the network, which leads to unbalanced traffic load over different links. Although the average delay is minimized in the network, but some switches may have doubled delay compared with other links. We try to overcome this shortcoming by defining and adding a new constraint per link to the main problem to make the delay lower than a threshold. This threshold will become the new objective function which we minimize. Solution time needed to solve the problem can be considered as another shortcoming where solving time has to be reduced as possible because all the packets related to the same flow has to wait until the control message arrive from the controller.

Hence, the controller need to solve the problem as fast as possible. We reformulate the problem in order to reduce the solution time.

3. System Model

In this section, network topology and traffic pattern are presented. In addition, parameters and variables used to describe the network model are defined.

A. *Network topology*

We consider the typical SDN contains of the hybrid Open flow empowered switches (like OF switches), that determined of the plane of data, the centralized network controller, control plane like it demonstrated in Figure 1. The SDN is able to be formed with the graph of network G = (V, J), that V is the switches set by whole of the n switches (i.e., |V| = n), J is the links set by whole of the links of |J| (Figure 2). An In-band mode is studied where control traffic is serviced via main links of network.

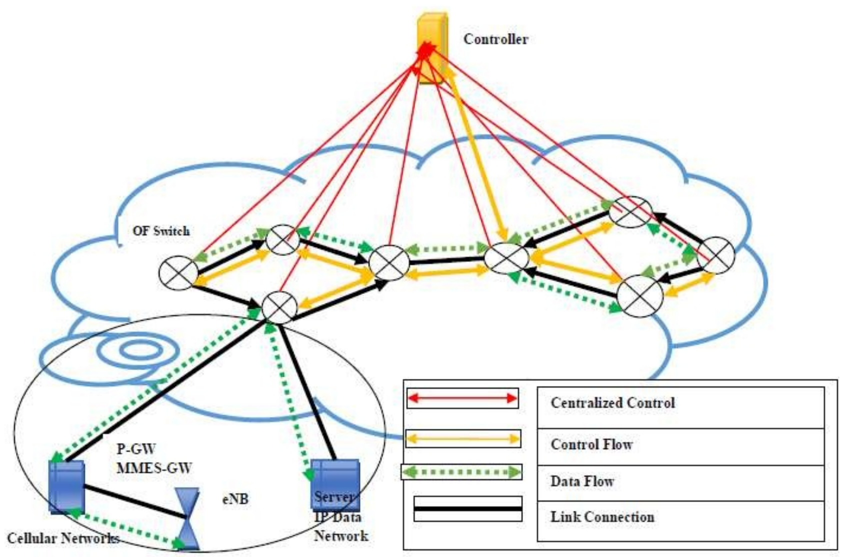

Figure 1. Network Architecture.

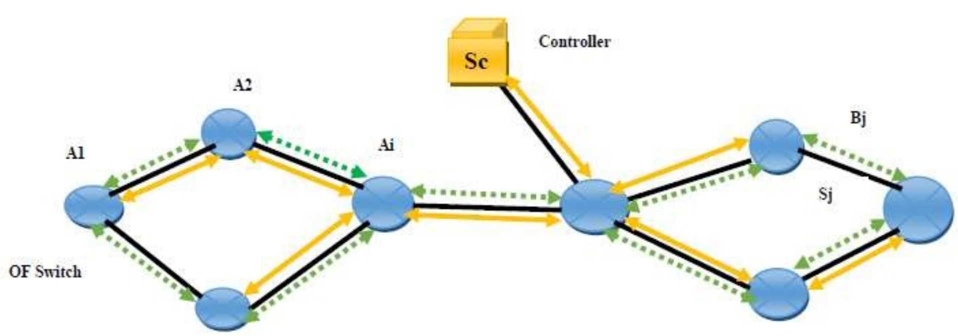

Figure 2. Network Topology as a Graph

When the novel flow comes to one switch of OF, the switch must send the message of control to central controller for asking that a route which is optimal. When controller of SDN possess the perfect overview on network, this is able to reply switch of OF better effective. So, controller optimizes optimal routes to every data flow in order to the actively alternating samples of traffic, flow needs of QoS, installs switches routing tables over best path through the protocols which are surely secure (like, Openflow) arrived control messages-based continuously.

B. *Pattern of traffic*

In our design, we consider two flows of data, control as the renovator progress. Particularly, every switch i control traffic formed with the renovator entrance process Ai by concept of the amount σi. To link of jth, j∈J, existence data flow pursues the renovator entrance progress Bj by concept of amount λj, link servicing time Sj pursues the other renovator progress by meanwhile 1/μj. In addition, centralized controller i* that is optimal, has the ability of serving by meanwhile 1/μc

4. Traffic Balancing Problem Control

In this section, we define the design of load balancing which optimize the link traffic loads in order to extra control traffic to the in-band control. This is supposed that central controller of SDN is able to hand over total flows of data, control. These architecture of network is accepted successfully in the true SDNs development. First, we define parameters and variables for the load balancing optimization problem in general. Then, we introduce the model presented in [1], and finally, we introduce our changes which leads to more load balancing over the links and reducing solving time of the problem.

A. Balancing problem's Formulation

First, we define the variables x_{ij} as control traffic value which i^{th} switch joins to link of J^{th}. We are able to explain the variables in matrix of as$\mathbf{x} = \left[x_{\ ij} \right], i \in$ V, j ∈ J. To every multi-path routing accepted in that P_i is the accessible paths set to i^{th} switch, $i \in$ V to central controller. It concepts that the switch i is able to transfer messages of control to controller through the $|P_i|$ accessible paths. So, to every switch i ,we explain the topology matrix T_i in size $|J| \times |P_i|$ like below:

$$T_i\ [j,p] = \begin{cases} 1\ , if\ the\ j^{th}\ link\ lies\ on\ the\ p^{th}\ path; \\ 0\ , otherwise. \end{cases} \qquad (1)$$

Matrix T_i maps traffic of paths to the links. The matrix T_i must be full column row always for preventing the random paths. This is left reverse matrix $T_i^{-1} = [t_{i1}, t_{i2}, \ldots, t_{i\ |J|}\,]$ that exists, possess size of $|P_i| \times |J|$, that t_{ij} is vector of column which maps J^{th} link to whole of the feasible paths of i^{th} switch's flow. t_{ij} got with multi-plying T_i^{-1} by standard of J^{th} on basic e_j , $i.e$, $t_{ij} = T_i^{-1} e_{\ j}$.

Since every switch $\boldsymbol{i}$ produces the control flow by concept amount $\boldsymbol{\sigma_i}$, switch $\boldsymbol{i}^*$, where controller straightly joined, is able to send the flow toward the controller with no going via network $\left(\boldsymbol{i.e}, \boldsymbol{x}_{\ i^*j} = \mathbf{0}\,, \forall\, \boldsymbol{j} \in \boldsymbol{J}\right)$. We install equalizes to switches control flow protection as$\|\boldsymbol{T_i^{-1}}\ [\boldsymbol{x_{i1}}, \ldots, \boldsymbol{x_{(i|J|)}}\ \ ]^{\boldsymbol{T}}\ \|_{\boldsymbol{1}}=\ , \forall\, \boldsymbol{i} \in \widetilde{\boldsymbol{V}}\ := \boldsymbol{V} \setminus \{\, \boldsymbol{i}^* \}$, where $\|.\|^{\boldsymbol{T}}$, $\|.\|_{\boldsymbol{1}}$ determine vector of 1-norm, transpose, sequentially. **Let** $\boldsymbol{d_{ij}} = \|\boldsymbol{T_i^{-1}} \boldsymbol{e_j}\, \|_{\boldsymbol{1}}$, the conservation constraint is able to be set as below:

$$\sum_{j=J} \boldsymbol{d_{ij}x_{ij=\sigma_i}} \ \forall \boldsymbol{i} \in \widetilde{\boldsymbol{V}} \qquad (\mathbf{2})$$

The constraint explains that control flow triggered with every switch i separate in hybrid out-going flows on chosen transferring links, however hybrid flows summation are equivalent to σ_i, always. Furthermore, utilizing little's rule, middle delay of network D across network to messages of control achieved as:

$$D = \frac{1}{\sum_{i\epsilon v} \sigma_i + \sum_{j\in J} \lambda_j} \sum_{j\epsilon J} \frac{\sum_{i\epsilon \widetilde{V}} x_{ij} + \lambda_j}{\mu_j - \left(\sum_{i\epsilon \widetilde{V}} x_{ij} + \lambda_j\right)} \qquad (3)$$

Particularly, to the link j ∈ J, novel packages come by the rate of ($\sum_{i\epsilon \widetilde{V}} x_{ij} + \lambda_j$), remain the middle time of 1 / [$\mu_j - \left(\sum_{i\epsilon \widetilde{V}} x_{ij} + \lambda_j\right)$]. Accumulating the backlogs of row along whole of the links, middle delay of network is so gave, as whole control, data traffic external entrance in network are($\sum_{i\epsilon v} \sigma_i + \sum_{j\in J} \lambda_j$). In fact, for balancing traffic loads between whole of the links, each link must possess the limited delay of transferring. From formulation in (3), these limited delay of link situations are equal to:

$$\sum_{i\epsilon \widetilde{V}} x_{ij} < \mu_j - \lambda_j \, , \ \forall_j \epsilon J \qquad (4)$$

B. Control Traffic Load-Balancing Problem

Description: The issue of Control Traffic Load-Balancing Given the SDN schemed with $G = (V, J)$ by controller state $i^* \epsilon V$, rates of control traffic entrance σ_i, the topology matrices set $T_i, \forall_{i \epsilon V}$, rates of data traffic λ_i, rates of link servicing μ_j, ∀ j ∈ J, issue of load- balancing optimization for being solved with controller is:

$$\text{Minimize D}\left(x_{ij:} \ {}_{\forall_{j\epsilon J}}^{\forall_{i\epsilon \widetilde{V}}} = V \setminus \{i^*\}\right)$$

$$\text{Subject to:} \quad 1. \ \sum_{j=J} d_{ij}x_{ij} \ \ \sigma_i \ \forall i \in \widetilde{V}, \qquad (5)$$

$$2. \sum_{i\epsilon \widetilde{V}} x_{ij} < \mu_j - \lambda_j \, , \ \forall_j \epsilon J$$

C. Discussion and Reformulating the problem

As we can see, the objective function represents the average network delay which is aimed to be minimized. We start from this point to change D which represents the average total delay to represent the delay for each link $j \in J$ that can be – based on Little's law- given as

$$\frac{1}{\sum_{i\epsilon_{\widetilde{V}}} \sigma_i + \lambda_j} \cdot \frac{\sum_{i\epsilon_{\widetilde{V}}} x_{ij} + \lambda_j}{\mu_j - \left(\sum_{i\epsilon_{\widetilde{V}}} x_{ij} + \lambda_j\right)} \qquad (6)$$

which represents the delay for link $j \in J$ the objective.

Now, instead of minimizing the average network delay, we add the Eq. (6) to the optimization problem's constraints to guarantee that the control traffic will be assigned to the links in such a way that each link will have a delay no more than D'

$$\frac{1}{\sum_{i\epsilon\tilde{V}} \sigma_i \ + \ \lambda_j} \cdot \frac{\sum_{i\epsilon\tilde{V}} x_{ij} \ + \ \lambda_j}{\mu_j - \left(\sum_{i\epsilon\tilde{V}} x_{ij} \ + \ \lambda_j\right)} \leq D' \qquad (7)$$

Obviously, with small values of D', the delay of each links becomes lower. However, when D' becomes smaller than a threshold, the problem becomes infeasible. Therefore, we need to choose the smallest value of D' under which the problem remains feasible. From this point, we define D' as our objective function which we want to minimize under the constraints (2), (4), and (6). Form the above discussion; we define our optimizing problem as the follow:

Minimize D'
Subject to:

1. $\frac{1}{\sum_{i\epsilon\tilde{V}} \sigma_i \ + \lambda_j} \cdot \frac{\sum_{i\epsilon\tilde{V}} x_{ij} \ + \lambda_j}{\mu_j - \left(\sum_{i\epsilon\tilde{V}} x_{ij} \ + \lambda_j\right)} \leq \quad D' \quad , \quad \forall j \in J$
2. $\sum_{j=J} d_{ij} x_{ij= \sigma_i} \ \forall \ i \ \in \tilde{V},$ (8)
3. $\sum_{i\epsilon\tilde{V}} x_{ij} \ < \ \mu_j - \lambda_j \ , \ \forall j \in J$
4. $x_{i,j} \geq 0; \forall \ i \ \in V, \forall j \in J.$

Compared with problem (4), we can see that the complexity of calculating the objective function is decreased dramatically, where in (4) the objective function represents the average network delay which is a non-linear expression. In (7), the objective function is linear. On the other side, we have $|J|$ non-linear constraints in problem (7). However, adding more constraints to the problem reduces the solution space and can lead to a more optimal solution in relatively shorter time.

5. Performance Evaluation

In our simulation, we consider the Internet2 OS3E scenario [24] as shown in Figure 3. Internet2 OS3E has 27 nodes and 36 links. In some scenarios, we increase number of links to see how complexity and solving time change with large-scale network sizes. We consider the data arrival rate λ_i=200 *packets/slot*, and serving rate for each link $j\in$ is 1000 *packets/slot*. The control traffic arrival rate is σ_i =40 *packets/slot*, but we change this rate in order to evaluate the optimization problems under different conditions. MATLAB is used to implement the simulation because the optimization problems (4), and (7) are complex to implement using other simulators. The simulation because the optimization problems.

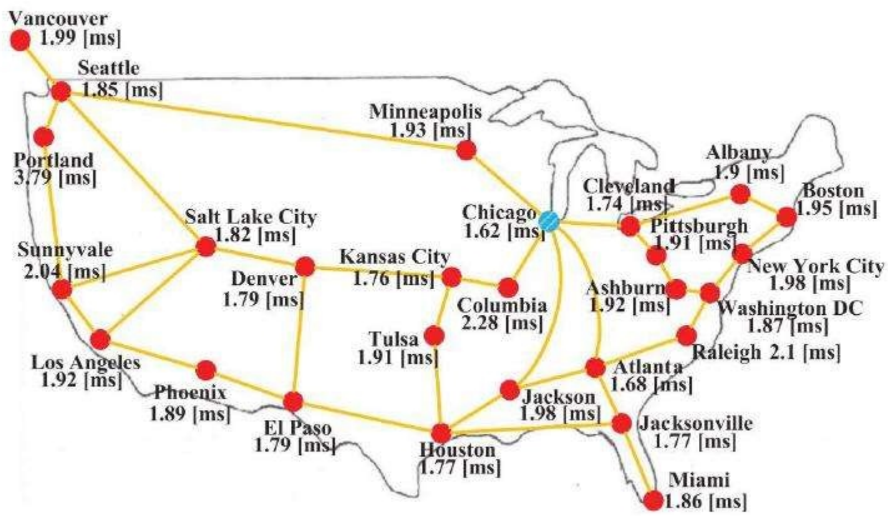

Figure 3. Internet2 OS3E

A. Improving Control Traffic Balancing

Since problem (4) –refer to as PTAA- aims to minimize the average network delay, the average network delay will be minimized, but with unbalanced control traffic loads on different links.

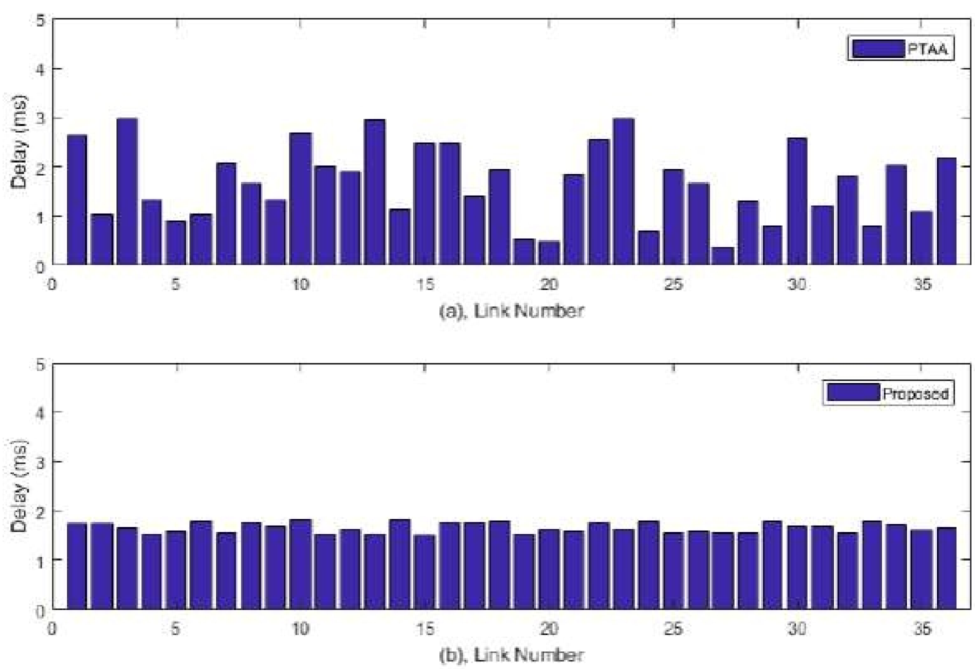

Figure 4. Balancing control traffic loads over network links

As shown in Figure 4 (a), even though the problem (4) minimizes the average network delay, it assigns high unbalanced delay levels to the network links. This will lead to different levels of

delays for different links and paths. Therefore, the constrain (6) in the optimization problem (7) adds more harmonies to the links' delay. As you can see in Figure 4 (b), differences in delay levels between network links become smaller. Therefore, control traffic in problem (7) is assigned more fairness.

B. Solving Time

Changing the objective delay from representing the average network delay to represent the delay per link, can achieve more fairness in between links in the network. In addition, removing the delay from the objective function and adding it to constrain can decrease the solution space. Decreasing solution space can make the operation of finding optimal solution (or near optimal solution) faster [25]. As depicted in Figure 5, solving time for problem (7) decreases to about half compared with problem (4) with 36 links. By increasing number of links, complexity of problems increases, and hence, solving time increasing accordingly. As we can see in the figure, by increasing number of links from 36 to 100, solving time for the problem (4) increases from 2.1 to 4.5, while solving time for problem (7) increases from 1 to less than 2, which reflect more scalability for our proposed scheme.

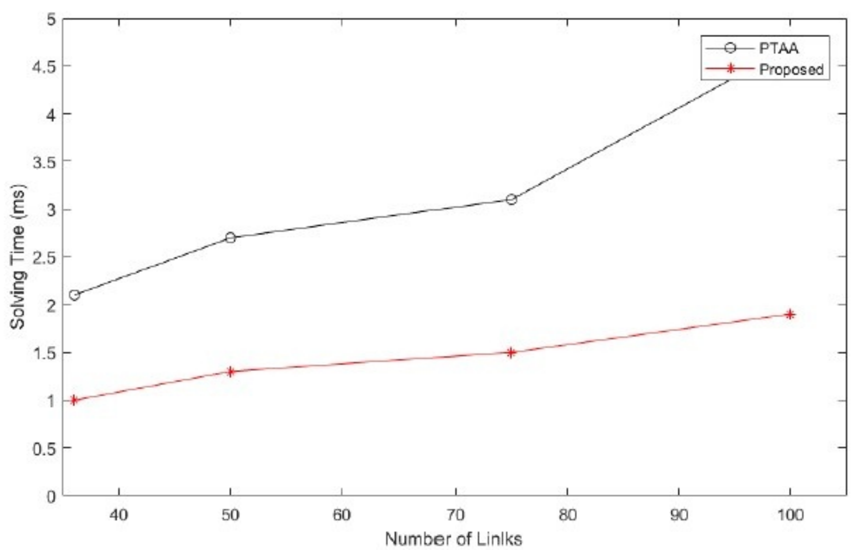

Figure 5. Solving time with respect to number of links.

C. Average Network delay

Now, after showing that problem (7) can assign more fairness control traffic to the network links, and it can be solved with less time, we need to check now its optimality. Since both problems (4) and (7) are nonconvex problems, it cannot be claim that one of them can reach optimal solution. Therefore, we compare the problem with each other to show the optimality.

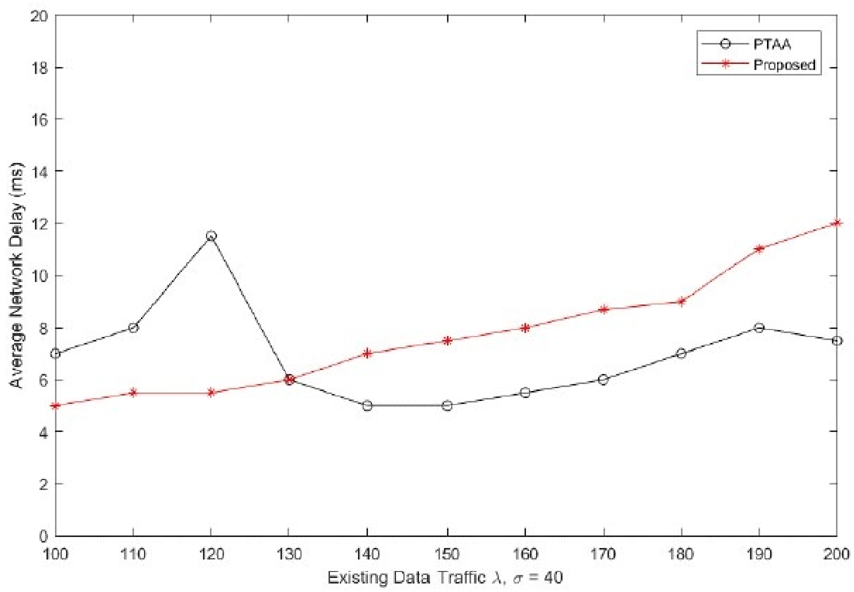

Figure 6. Average Network Delay with respect to

We evaluate here average network delay with respect to data traffic arrival rates, where i increases from 100 to 200 $packets/slot$. As shown in Figure 6. Problem (4) is not able to achieve stable solutions with respect to λi. When λi=110, the average network delay is about $8m$ while increases to about $12m$ with λi =120. Unexpectedly, the average network delay decreases again to about $6m$ with λi=130. The behavior of the problem (4) shows that for many values of λi, it gives non-near optimal solution, because in practical, the average network delay for λi=130 has to be more than delay for λi=120. On the other side, Figure 6. Shows that, in problem (7) the average network delay gradually increases with λi. However, with some values of λi, problem (4) can give less average delay in the network. This is expected because the main aim of problem (4) is to minimize the average network delay, while problem (7) aims to minimize the delay for each link. In the same way, we evaluate the average network delay with respect to control arrival rate σi. Similarly, problem (4) shows un-stability for increasing σi while it can achieve less average delay than problem (7).

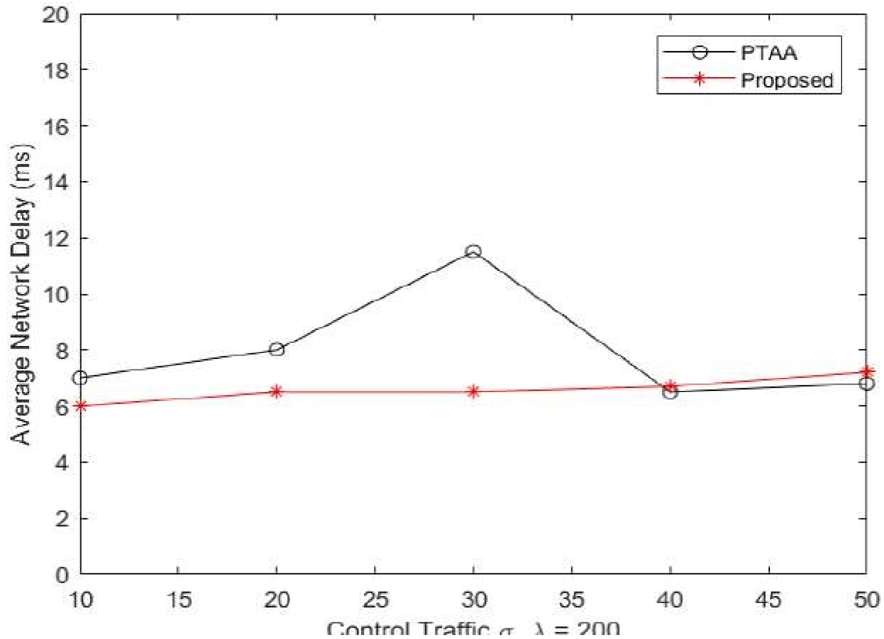

Figure 7. Average Network Delay with respect to σ

D. Total Response time

When we partially observe figures 6 and 7, we can see that the problem (4) can achieve less average delay than problem (7). This will give a negative evaluation for the problem. However, this comparison is unfair and based on partial evaluation. Generally, when the central controller receives a request from one switch, it needs to solve a problem (4 or 7), then using the best route it found, it responses to the requested switch. Therefore, the total response time for the requested switch equals to the solving time needed to solve the problem in addition to the delay from central controller to the switch. Therefore, the figures 8, and 9 are presented to show the average network delay considering solving time duration with respect to data traffic arrival rate λi, and control traffic arrival rate σi.

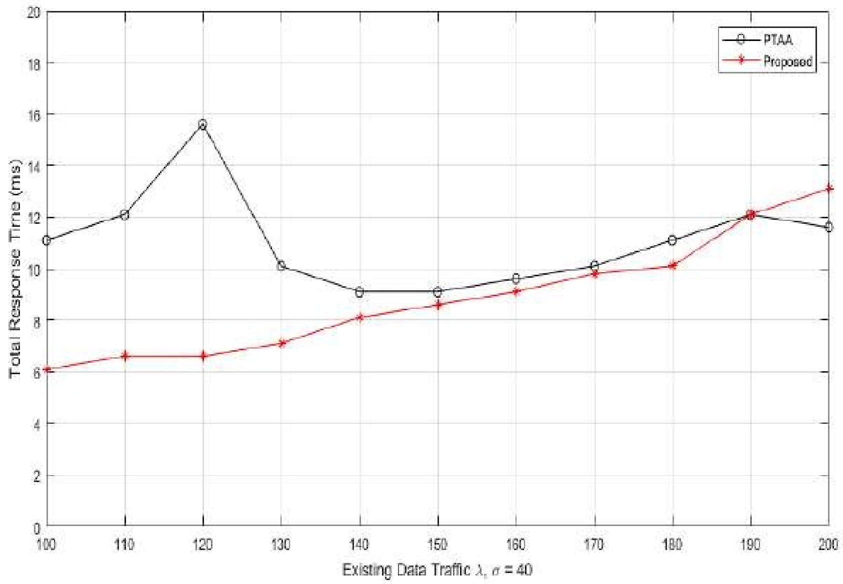

Figure 8. Total response Time with respect to

Figure 8 shows that problem (7) can achieve less response time than problem (4) in most cases when the data traffic arrival rate increases from 100 to 200. In the same way, the total response time achieved by problem (7) is always less than total response time achieve by problem (4).

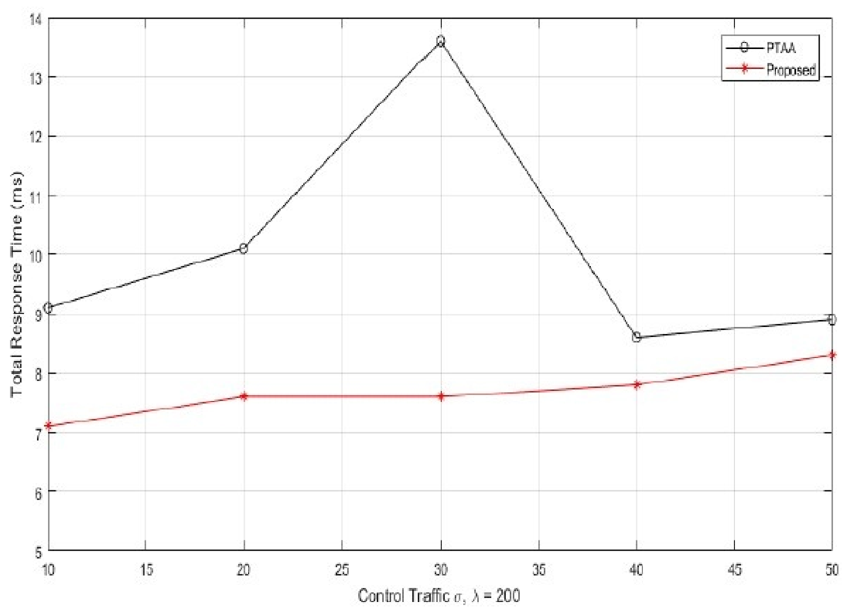

Figure 9. Total response Time with respect to σ.

6. Conclusion

During this work, a definition of software defined networking has been introduced. Related works showed that this network architecture will play a major role in the future networks architectures. Traffic load balancing in this networks one of the well-known problems which received high attention over the recent years by the researchers. While many works studied data traffic load balancing and ignored control traffic, we have introduced one recent work that tried to consider control traffic together with data traffic. Even though this work presented the control traffic as a big problem for software defined networking, it had many shortages and defects. In this work, we have tried to overcome these shortages and defects, and our simulation results shows that more load balancing can be achieved in our proposed scheme. In addition, the optimization problem which is needed to be solved online to determine the optimal load over each links is reformulated in which that can be solved in shorter time.

References

[1] Lin S, Wang P, Luo M. Control traffic balancing in software defined networks. Computer Networks 2016; 106: 260-271.

[2] McKeown N, Anderson T, Balakrishnan H, Parulkar G, Peterson L, Rexford J, Shenker S, Turner J. Openflow: enabling innovation in campus networks, ACM SIGCOMM. Computer Communication Review 2008; 38(2): 69–74.

[3] Akyildiz I, Lee A, Wang P, Luo M, Chou W. A roadmap for traffic engineering in sdn-openflow networks. Computer Networks 2014; 71: 1–30.

[4] Akyildiz I, Wang P, Lin S. Softair: a software defined networking architecture for 5g wireless systems Computer Networks 2015; 85:1–18.

[5] ONF. OpenFlow Switch Specification. Version 1.4.0. https://www.opennetworking.org/images/stories/downloads/sdnres

[6] Yeganeh S, Tootoonchian A, Ganjali Y. On scalability of software-defined networking, IEEE Communications Magazine 2013; 51(2): 136–141.

[7] Lin S, Wang P, Luo M. Control traffic balancing in software defined networks, Computer Networks. 2016; 106: 260-271.

[8] Ropke C, Holz T. Retaining Control Over SDN Network Services. International Conference on Networked Systems 2015; 1: 1-5.

[9] Handigol N., Flajslik M, Seetharaman S. Aster*x: Load-Balancing as a Network Primitive. 9th GENI Engineering Conference 2010; p1-2.

[10] Zerrik S, Bakhouya M, Gaber J. Towards a Decentralized and Adaptive Software Defined Networking Architecture. Fifth International Conference on Next Generation Networks and Services, 2014.

[11] Braun W, Menth M. Software-Defined Networking Using openflow: Protocols, Applications and Architectural Design Choices, Future Internet 2014.

[12] Software-Defined Networking: The New Norm for Networks, Open Networking Foundation White Paper, April 2012.

[13] Koerner M, Kao O. Multiple Service Load-Balancing with openflow. High Performance Switching and Routing 13th International Conference 2012.

[14] Shang Z, Chen W, Ma Q, WU B. Design and implementation of server cluster dynamic load balancing based on openflow. Awareness Science and Technology and Ubi –Media Computing, International Joint Conference 2013.

[15] Zhang H, Guo X. SDN-based load balancing strategy for server cluster. Proceedings of CCIS 2014.

[16] Saifullah M, Mohamed M. Open Flowbased Server Load Balancing using Improved Server Health Reports. International Conference on Advances in Electrical, Electronics, Information, Communication and Bio Informatics 2016.

[17] Chin M, Tan C, Bandan M. Efficient Load Balancing for Bursty Demand in Web based Application Services via Domain Name Services. International Journal of Science the and Internet 2012; 1(1).

[18] Wadhwa D, Kumar N. Performance Analysis of Load Balancing Algorithms in Distributed System. Advance in Electronic and Electric Engineering 2014; 4 (1): 59-66.

[19] Yong W, Xiaoling T, HE Qian, Kuang Y. A Dynamic Load Balancing Method of Cloud-Center Based on SDN. China Communications, 2016.

[20] Long H, Shen Y, Guo M, Tang F. LABERIO: Dynamic loadbalanced routing in OpenFlow-enabled networks. 2013 IEEE 27th International Conference on Advanced Information Networking and Applications 2013.

[21] Chandra K. Performance Analysis of Load Balancing Algorithms for cluster of Video on Demand Servers 2009 WEJEJ International Advance Computing Conference Patiala, India, 6-7 March 2009.

[22] Zhou W, Yang S, Fang J, Niu X, Song Hu. vmctune: A Load Balancing Scheme for Virtual Machine Cluster Based on Dynamic Resource Allocation. Ninth International Conference on Grid and Cloud Computing 2010.

[23] Liu J, Dong F. A Dynamic Adaptive Load Balance Algorithm in Parallel Intrusion Detection System. International Symposium on Computer Science and Computational Technology 2008.

[24] Koponen T, Casado M, Gude N, Stribling J, Poutievski L, Zhu M, Ramanathan R, Iwata Y, Inoue H, Hama T, Shenker S. Onix: A Distributed Control Platform for Large-Scale Production Networks, Proc. 9th USENIX.

[25] Feillet D. A tutorial on column generation and branch-and-price for vehicle routing problems. 4OR-Q Operational Research 2010; 8: 407–424. https://doi.org/10.1007/s10288-010-0130-z